Analisis y Diseño de Algoritmos

Juan Miguel Rojas

5 de febrero de 2024

1. Código de honor

Como miembro de la comunidad académica de la Pontificia Universidad Javeriana Cali, los valores éticos y la integridad son tan importantes como la excelencia académica. En este curso se espera que los estudiantes se comporten ética y honestamente, con los más altos niveles de integridad escolar. En particular, se asume que cada estudiante adopta el siguiente código de honor:

Como miembro de la comunidad académica de la Pontificia Universidad Javeriana Cali me comprometo a seguir los más altos estándares de integridad académica.

Integridad académica se refiere a ser honesto, dar crédito a quien lo merece y respetar el trabajo de los demás. Por eso es importante evitar plagiar, engañar, 'hacer trampa', etc. En particular, el acto de entregar un programa de computador ajeno como propio constituye un acto de plagio; cambiar el nombre de las variables, agregar o eliminar comentarios y reorganizar comandos no cambia el hecho de que se está copiando el programa de alguien más. Para más detalles consultar el Reglamento de Estudiantes, Sección VI.

2. Comparison of running times

For each function $f(n)$ and time t in the following table, determine the largest size n of a problem that can be solved in time t, assuming that the algorithm to solve the problem takes $f(n)$ microseconds.

	1 second	1 minute	1 hour	1 day	1 month	1 year	1 century
$\lg n$							
$\sqrt{n}$							
n							
$n \lg n$							
n^2							
n^3							
2^n							
$n!$							

Figura 1: Comparison of running times

Calcular n

En mi planteamiento tomaremos como caso base calcular **n**, ya que este depende para calcular las demas complejidades. Entonces esta será la inicial.

- **1 second**
  Tenemos que un segundo es igual a 10^6, por lo tanto nuestro **n** para 1 second es 10^6.
- **1 minute**
  Tenemos que un segundo es igual 10^6, por lo tanto un minuto son 60 segundos y de este modo lo calculamos $60 \cdot 10^6 = 6 \cdot 10^7$.
- **1 hour**
  Tenemos que un minuto es igual $6 \cdot 10^7$, por lo tanto una hora son 60 minutos y de este modo lo calculamos $60 \cdot (6 \cdot 10^7) = 3{,}6 \cdot 10^9$.
- **1 day**
  Tenemos que una hora es igual $3{,}6 \cdot 10^9$, por lo tanto un dia son 24 horas y de este modo lo calculamos $24 \cdot (3{,}6 \cdot 10^9) = 86{,}4 \cdot 10^9$.
- **1 month**
  Tenemos que un dia es igual $86{,}4 \cdot 10^9$, por lo tanto un mes son 30 dias y de este modo lo calculamos $30 \cdot (86{,}4 \cdot 10^9) = 2592 \cdot 10^9$.
- **1 year**
  Tenemos que un mes es igual $259{,}2 \cdot 10^9$, por lo tanto un años son 12 meses y de este modo lo calculamos $12 \cdot (259{,}2 \cdot 10^9) = 3{,}1104 \cdot 10^{13}$.
- **1 century**
  Tenemos que un año es igual $3{,}1104 \cdot 10^{13}$, por lo tanto un siglo son 100 años y de este modo lo calculamos $100 \cdot (3{,}1104 \cdot 10^{13}) = 3{,}1104 \cdot 10^{15}$

Calcular n^2

En este caso partimos de nuestro caso base n, con el cual cada valor lo elevaremos al cuadrado y obtendremos su resultado, pero como necesitamos calcular n para n^2, tomaremos la inversa de la función.

- **1 second**
  Tenemos que para n un segundo es igual a 10^6, por lo tanto hacemos el calculo de $\sqrt{10^6} = 1000$.
- **1 minute**
  Tenemos que para n un minuto es igual a $6 \cdot 10^7$, por lo tanto hacemos el calculo de $\sqrt{6 \cdot 10^7} = 2000 \cdot \sqrt{15}$ ó 7745.96.
- **1 hour**
  Tenemos que para n una hora es igual a $3{,}6 \cdot 10^9$, por lo tanto hacemos el calculo de $\sqrt{3{,}6 \cdot 10^9} = 60000$.
- **1 day**
  Tenemos que para n un dia es igual a $86{,}4 \cdot 10^9$, por lo tanto hacemos el calculo de $\sqrt{86{,}4 \cdot 10^9} = 120000 \cdot \sqrt{6}$ ó 293938.77.

- **1 month**
  Tenemos que para n un mes es igual a $259{,}2 \cdot 10^9$, por lo tanto hacemos el calculo de $\sqrt{2592 \cdot 10^9} = 720000 \cdot \sqrt{5}$ ó 1609968.94.

- **1 year**
  Tenemos que para n un año es igual a $3{,}1104 \cdot 10^{13}$, por lo tanto hacemos el calculo de $\sqrt{3{,}1104 \cdot 10^{13}} = 1440000 \cdot \sqrt{15}$ ó 5577096.01.

- **1 century**
  Tenemos que para n un siglo es igual a $3{,}1104 \cdot 10^{15}$, por lo tanto hacemos el calculo de $\sqrt{3{,}1104 \cdot 10^{15}} = 14400000 \cdot \sqrt{15}$ ó 55770960.18.

Calcular n^3

Utilizamos la misma dinamica, partimos de nuestro caso base n y calculamos el resultado de n^3 utilizando la inversa de esta función.

- **1 second**
  Tenemos que para n un segundo es igual a 10^6, por lo tanto hacemos el calculo de $\sqrt[3]{10^6} = 100$.

- **1 minute**
  Tenemos que para n un minuto es igual a $6 \cdot 10^7$, por lo tanto hacemos el calculo de $\sqrt[3]{6 \cdot 10^7} = 100 \cdot \sqrt[3]{60}$ ó 391.48.

- **1 hour**
  Tenemos que para n una hora es igual a $3{,}6 \cdot 10^9$, por lo tanto hacemos el calculo de $\sqrt[3]{3{,}6 \cdot 10^9} = 1532{,}61$.

- **1 day**
  Tenemos que para n un dia es igual a $86{,}4 \cdot 10^9$, por lo tanto hacemos el calculo de $\sqrt[3]{86{,}4 \cdot 10^9} = 4420{,}83$.

- **1 month**
  Tenemos que para n un mes es igual a $259{,}2 \cdot 10^9$, por lo tanto hacemos el calculo de $\sqrt[3]{2592 \cdot 10^9} = 13736{,}57$.

- **1 year**
  Tenemos que para n un año es igual a $3{,}1104 \cdot 10^{13}$, por lo tanto hacemos el calculo de $\sqrt[3]{3{,}1104 \cdot 10^{13}} = 31448{,}89$.

- **1 century**
  Tenemos que para n un siglo es igual a $3{,}1104 \cdot 10^{15}$, por lo tanto hacemos el calculo de $\sqrt[3]{3{,}1104 \cdot 10^{15}} = 145972{,}84$.

Calcular $\sqrt{n}$

Utilizamos la misma dinamica, partimos de nuestro caso base n y calculamos el resultado de $\sqrt{n}$ utilizando la inversa de esta función.

- **1 second**
  Tenemos que para n un segundo es igual a 10^6, por lo tanto hacemos el calculo de $(10^6)^2 = 10^{12}$.

- **1 minute**
  Tenemos que para n un minuto es igual a $6 \cdot 10^7$, por lo tanto hacemos el calculo de $(6 \cdot 10^7)^2 = 36 \cdot 10^{14}$.

- **1 hour**
  Tenemos que para n una hora es igual a $3{,}6 \cdot 10^9$, por lo tanto hacemos el calculo de $(3{,}6 \cdot 10^9)^2 = 12{,}96 \cdot 10^{18}$.

- **1 day**
  Tenemos que para n un dia es igual a $86{,}4 \cdot 10^9$, por lo tanto hacemos el calculo de $(86{,}4 \cdot 10^9)^2 = 7464{,}96 \cdot 10^{18}$.

- **1 month**
  Tenemos que para n un mes es igual a $259{,}2 \cdot 10^9$, por lo tanto hacemos el calculo de $(2592 \cdot 10^9)^2 = 6{,}71846 \cdot 10^{24}$.

- **1 year**
  Tenemos que para n un año es igual a $3{,}1104 \cdot 10^{13}$, por lo tanto hacemos el calculo de $(3{,}1104 \cdot 10^{13})^2 = 9{,}67458816 \cdot 10^{26}$.

- **1 century**
  Tenemos que para n un siglo es igual a $3{,}1104 \cdot 10^{15}$, por lo tanto hacemos el calculo de $(3{,}1104 \cdot 10^{15})^2 = 9{,}67458816 \cdot 10^{30}$.

Calcular 2^n

Utilizamos la misma dinamica, partimos de nuestro caso base n y calculamos el resultado de 2^n utilizando la inversa de esta función.

- **1 second**
  Tenemos que para n un segundo es igual a 10^6, por lo tanto hacemos el calculo de $\log_2(10^6) = 19{,}93$.

- **1 minute**
  Tenemos que para n un minuto es igual a $6 \cdot 10^7$, por lo tanto hacemos el calculo de $\log_2(6 \cdot 10^7) = 25{,}83$.

- **1 hour**
  Tenemos que para n una hora es igual a $3{,}6 \cdot 10^9$, por lo tanto hacemos el calculo de $\log_2(3{,}6 \cdot 10^9) = 32{,}5$.

- **1 day**
  Tenemos que para n un dia es igual a $86{,}4 \cdot 10^9$, por lo tanto hacemos el calculo de $\log_2(86{,}4 \cdot 10^9) = 36{,}5$.

- **1 month**
  Tenemos que para n un mes es igual a $259{,}2 \cdot 10^9$, por lo tanto hacemos el calculo de $\log_2(2592 \cdot 10^9) = 41{,}24$.

- **1 year**
  Tenemos que para n un año es igual a $3{,}1104 \cdot 10^{13}$, por lo tanto hacemos el calculo de $\log_2(3{,}1104 \cdot 10^{13}) = 44{,}82$.

- **1 century**
  Tenemos que para n un siglo es igual a $3{,}1104 \cdot 10^{15}$, por lo tanto hacemos el calculo de $\log_2(3{,}1104 \cdot 10^{15}) = 51$.

Calcular $\log n$

Utilizamos la misma dinamica, partimos de nuestro caso base n y calculamos el resultado de $\log n$ utilizando la inversa de esta función.

- **1 second**
  Tenemos que para n un segundo es igual a 10^6, por lo tanto hacemos el calculo de $2^{(10^6)} = 2^{1000000}$.

- **1 minute**
  Tenemos que para n un minuto es igual a $6 \cdot 10^7$, por lo tanto hacemos el calculo de $2^{(6 \cdot 10^7)} = 2^{60000000}$.

- **1 hour**
  Tenemos que para n una hora es igual a $3{,}6 \cdot 10^9$, por lo tanto hacemos el calculo de $2^{(3{,}6 \cdot 10^9)} = 2^{3600000000}$.

- **1 day**
  Tenemos que para n un dia es igual a $86{,}4 \cdot 10^9$, por lo tanto hacemos el calculo de $2^{(86{,}4 \cdot 10^9)} = 2^{(86{,}4 \cdot 10^9)}$.

- **1 month**
  Tenemos que para n un mes es igual a $2592 \cdot 10^9$, por lo tanto hacemos el calculo de $2^{(2592 \cdot 10^9)} = 2^{2592000000000}$.

- **1 year**
  Tenemos que para n un año es igual a $3{,}1104 \cdot 10^{13}$, por lo tanto hacemos el calculo de $2^{(3{,}1104 \cdot 10^{13})} = 8{,}63622\ldots^{10^{13}}$.

- **1 century**
  Tenemos que para n un siglo es igual a $3{,}1104 \cdot 10^{15}$, por lo tanto hacemos el calculo de $2^{(3{,}1104 \cdot 10^{15})} = 8{,}63622\ldots^{10^{15}}$.

Calcular $n \cdot \log n$

Utilizamos la misma dinamica, partimos de nuestro caso base n y calculamos el resultado de $n \cdot \log n$ utilizando la inversa de esta función, pero en este caso tendremos que utilizar un pequeño algoritmo en python debido a que es dificil calcular el valor, así que tendremos que aproximarlo.

- **1 second**
  Tenemos que para n un segundo es igual a 10^6, por lo tanto hacemos el calculo con la función de python y nos da 87849.

- **1 minute**
  Tenemos que para n un minuto es igual a $6 \cdot 10^7$, por lo tanto hacemos el calculo con la función de python y nos da 3950159.

- **1 hour**
  Tenemos que para n una hora es igual a $3{,}6 \cdot 10^9$, por lo tanto hacemos el calculo con la función de python y nos da 188909176.

- **1 day**
  Tenemos que para n un dia es igual a $86{,}4 \cdot 10^9$, por lo tanto hacemos el calculo con la función de python y nos da 3911758541.

- **1 month**
  Tenemos que para n un mes es igual a $259{,}2 \cdot 10^9$, por lo tanto hacemos el calculo con la función de python y nos da 81470318032.

- **1 year**
  Tenemos que para n un año es igual a $3{,}1104 \cdot 10^{13}$, por lo tanto hacemos el calculo con la función de python y nos da 992457491131.

- **1 century**
  Tenemos que para n un siglo es igual a $3{,}1104 \cdot 10^{15}$, por lo tanto hacemos el calculo con la función de python y nos da 73132678011261.

Codigo para poder calcular $n \cdot \log n$

```python
import math

def n_log_n(microseg):
    n = 1
    flag = 0
    while flag == 0:
        tiempo = n * math.log(n)
        if tiempo > microseg:
            flag = 1
        n += 1
    return n
```

Calcular $n!$

Utilizamos la misma dinamica, partimos de nuestro caso base n y calculamos el resultado de $n!$ utilizando la inversa de esta función, pero en este caso tendremos que utilizar un pequeño algoritmo en python debido a que es dificil calcular el valor, así que tendremos que aproximarlo.

- **1 second**
  Tenemos que para n un segundo es igual a 10^6, por lo tanto hacemos el calculo con la función de python y nos da 10.

- **1 minute**
  Tenemos que para n un minuto es igual a $6 \cdot 10^7$, por lo tanto hacemos el calculo con la función de python y nos da 12.

- **1 hour**
  Tenemos que para n una hora es igual a $3{,}6 \cdot 10^9$, por lo tanto hacemos el calculo con la función de python y nos da 13.

- **1 day**
  Tenemos que para n un dia es igual a $86{,}4 \cdot 10^9$, por lo tanto hacemos el calculo con la función de python y nos da 15.

- **1 month**
  Tenemos que para n un mes es igual a $259{,}2 \cdot 10^9$, por lo tanto hacemos el calculo con la función de python y nos da 16.

- **1 year**
  Tenemos que para n un año es igual a $3{,}1104 \cdot 10^{13}$, por lo tanto hacemos el calculo con la función de python y nos da 18.

- **1 century**
  Tenemos que para n un siglo es igual a $3{,}1104 \cdot 10^{15}$, por lo tanto hacemos el calculo con la función de python y nos da 19.

Codigo para poder calcular $n!$

```python
import math

def n_factorial(microseg):
    n = 1
    flag = 0
    while flag == 0:
        tiempo = n * math.factorial(n)
        if tiempo > microseg:
            flag = 1
        n += 1
    return n
```